Spectrally Adaptive Channel-aware Unrolling Network for Compressed Sensing

Xiaoyang Wang[1], **Hongping Gan**[1*]

[1]School of Software, Northwestern Polytechnical University
Xi'an 710072, China
wxyang@mail.nwpu.edu.cn, ganhongping@nwpu.edu.cn

Abstract

Deep Unrolling Networks (DUNs) integrate classical optimization recovery problems in Compressed Sensing (CS) with sophisticated deep learning network architectures, leading to substantial breakthroughs. However, prevailing DUNs generally face challenges concerning solidified gradient descent step size strategies, inadequate feature extraction within the iterative stage and limited information interaction between iterative stages. To overcome these obstacles, we propose SCU-Net, a channel-focused unrolling network inspired by the renowned spectral projected gradient optimization algorithm. In particular, we tailore two pivotal components, Barzilai-Borwein-gradient Descent Optimizer (BBDO) and Channel-guided Cross-attention Reconstruction Module (CCRM), to collaboratively undertake the reconstruction task. BBDO leverages a gradient calculation strategy based on BB step size to enhance data fidelity optimization, while CCRM addresses the intricate mapping issue associated with sparse induction, encompassing customized functionalities from Adaptive Channel Interaction Layer (ACIL) and Spatially Augmented Channel-aware Unit (SACU). Among them, ACIL amalgamates convolution operations and channel attention mechanisms to achieve meticulous information screening alongside efficient feature enhancement. SACU introduces dual reinforcement variables to bolster information exchange across different iterative stages, coupled with the optimization of cross-attention to facilitate the modeling of long-distance dependencies. Extensive experiments in both image CS and magnetic resonance imaging exhibit that our SCU-Net manifests superior performance, surpassing state-of-the-art methods.

Introduction

CS (Duarte and Baraniuk 2012) stands as a cutting-edge signal sampling theory capable of data compression during signal acquisition by reconstructing signals from minimal measurements with high fidelity. By breaking the limits of the traditional Nyquist sampling theorem, CS has become a key paradigm in modern signal processing. Moreover, it has catalyzed paradigm shifts in practical applications across diverse domains, including magnetic resonance imaging (MRI) (Jalal et al. 2021), snapshot compressive imaging (Wang et al. 2023; Cheng et al. 2023), and hyperspectral compressed imaging (Dong et al. 2022), among others (Meng and Kabashima 2024; Zhang et al. 2023).

The sampling process of CS can be formulated as $\mathbf{y} = \mathbf{\Phi x}$, where $\mathbf{y} \in \mathbb{R}^M$ represents the measurements, $\mathbf{\Phi} \in \mathbb{R}^{M\times N}(M \ll N)$ denotes the sampling matrix and $\mathbf{x} \in \mathbb{R}^N$ is the original signal. The CS sampling rate τ is defined as the ratio of M to N. Due to M is substantially smaller than N, traditional CS methods address the ill-posed problem of recovering $\mathbf{x}$ from $\mathbf{y}$ by usually reformulating it as a generalized LASSO problem to seek an appropriate solution:

$$\min_{\mathbf{x}\in\mathbb{R}^N} \frac{1}{2}\|\mathbf{\Phi x} - \mathbf{y}\|_{\ell_2}^2 + \lambda\|\mathbf{Kx}\|_{\ell_1}, \tag{1}$$

where $\mathbf{K}$ denotes the sparse transformation matrix, and λ is a regularization parameter that modulates the trade-off between the fidelity term $\|\mathbf{\Phi x} - \mathbf{y}\|_{\ell_2}^2$ and the sparsity term of the solution. Given the prevalent non-sparse nature of signals within the time domain in practical scenarios, the original signal $\mathbf{x}$ can be effectively transformed into a designated transform domain utilizing the sparse transformation matrix, facilitating the derivation of its sparse representation.

Notably, representative traditional convex optimization methodologies for CS recovery comprise the iterative shrinkage thresholding algorithm (ISTA) (Beck and Teboulle 2009), approximate message passing algorithm (AMP) (Borgerding, Schniter, and Rangan 2017), among others (Boyd et al. 2011; Bioucas-Dias and Figueiredo 2007). Despite making significant contributions to CS and sparse signal recovery, these algorithms continue to grapple with the inherent constraints of fixed gradient descent step size. In contrast, the spectral projected gradient algorithm (van den Berg and Friedlander 2009) introduces a more versatile approach to govern signal sparsity modeling through the incorporation of the Barzilai-Borwein (BB) (Barzilai and Borwein 1988) step parameter, thus augmenting its capability to handle intricate signal structures. However, conventional algorithms encounter numerous hurdles in practical deployment, including the manual tuning of hyperparameters and elevated computational demands, hindering their utility in exigent real-time processing and versatility across diverse practical application domains.

With the robust advancement of deep learning (DL), it has emerged as a novel paradigm for addressing im-

*Corresponding author.

age inverse problems. Researchers have proposed pure DL-based reconstruction algorithms to effectively learn the direct mapping between measurements and the original signal, such as AutoBCS (Gan et al. 2023a), TCS-Net (Gan et al. 2023b), MTC-CSNet (Shen et al. 2024), MCFD-Net (Song et al. 2024). Nevertheless, these methods remain an enigmatic black-box model. Conversely, deep unrolling networks (DUNs) embed deep architectures into the iterative update process, which not only enables parameter-adaptive learning but also contributes to enhancing interpretability rooted in the realm of CS. However, contemporary DUNs still encounter difficulties in precisely reconstructing intricate details under limited sampling conditions, falling short of satisfying the requirement for superior signal reconstruction quality. The fundamental reasons for these hurdles in DUNs are as follows: 1) Solidification of gradient descent step size strategies: current DL-based methods commonly adopt a fixed paradigm for self-learning gradient descent step size, completely dissociating from traditional optimization methods. 2) Insufficiency in feature extraction during the iteration stage: the absence of robust feature coupling strategies in existing DUNs precipitates the inability to simultaneously obtain both local details and global structures of images, thereby weakening the ability to capture fine-grained data. 3) Inadequate information fusion between iteration stages: the importance and contribution of information from different stages vary for the final reconstruction result. However, existing DUNs often fail to adaptively balance and fuse this information generated from different stages, leading to loss and confusion of information.

To address these challenges, we propose SCU-Net, a novel deep unrolling network for CS recovery, inspired by the spectral projected gradient for ℓ_1 optimization algorithm (SPGL1). Particularly, we devise the BB-gradient Descent Optimizer (BBDO) and Channel-guided Cross-attention Reconstruction Module (CCRM) specifically to undertake the core reconstruction tasks. The former integrates a gradient calculation strategy based on the BB step size to improve data fidelity, while the latter is dedicated to handling complex mapping problems associated to sparse representation. Furthermore, CCRM comprises two specially tailored sub-modules: Adaptive Channel Interaction Layer (ACIL) and Spatially Augmented Channel-attention Unit (SACU), which collaborate to achieve fine-grained image reconstruction. Our main contributions are summarized as follows:

- We propose SCU-Net, which combines the advantages of the SPGL1 optimization algorithm and DL, achieving superior reconstruction quality and model interpretability.
- We customize BB-gradient Descent Optimizer, incorporating the BB strategy to update the gradient descent step size, enhancing the control ability of parameter adaptive learning in gradient descent and accelerating the convergence speed.
- We design Adaptive Channel Interaction Layer, which integrates convolution operations and channel attention mechanisms, effectively breaking through local attribute limitations, significantly improving spatial perception capabilities and achieving precise information screening and efficient feature enhancement.
- We tailore Spatially Augmented Channel-attention Unit, introducing dual reinforcement variables to strengthen information interaction between different iterative stages, and simultaneously optimizing the cross-attention mechanism, thereby promoting the modeling of long-distance dependency relationships.

Extensive experiments demonstrate that our SCU-Net outperforms the current SOTA methods both on image CS and CS-MRI while exhibiting cost-effective parameter usage.

Related Works

SPGL1 Solver. The spectral projected gradient algorithm (SPGL1), introduced by Berg and Friedlander (van den Berg and Friedlander 2009), serves to address ℓ_1-constrained optimization problems as illustrated in Eq. (1). In essence, it utilizes the Barzilai-Borwein (BB) spectral step length in the projected gradient algorithm, which entails executing a gradient descent step based on the BB search length during each iteration, subsequently projecting the iteration point onto the feasible set of the generalized LASSO problem:

$$\mathcal{P}_\lambda(\mathbf{c}) := \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{x}-\mathbf{c}\|_{\ell_2}^2 + \lambda\|\mathbf{K}\mathbf{x}\|_{\ell_1}, \qquad (2)$$

where $\mathbf{c}$ represents iteration output, $\mathcal{P}_\lambda(\cdot)$ denotes the operation of projecting vector $\mathbf{c}$ onto the constraints. Refer to *Sect. 1 of* ***Supplementary Material (Supp)*** for more details.

Deep Unrolling Networks. Deep unrolling networks (DUNs) embed traditional iterative optimization algorithms into neural network forms typically composed of convolutional neural networks, attention mechanisms, or Transformers. This adaptation maintains the interpretability of the optimization algorithm while harnessing the benefits of DL to enhance efficiency and performance. Several extensively referenced classic examples, such as ISTA-Net (Zhang and Ghanem 2018) and AMP-Net (Zhang et al. 2020), have become benchmarks in the field of image CS. In particular, ISTA-Net has not only gained significant popularity but also spurred the emergence of a range of DUNs. For example, Shen et al. proposed TransCS (Shen et al. 2022), a hybrid deep architecture, which combines a customized ISTA-based Transformer backbone and convolutional neural networks to achieve high-quality image CS. Alternatively, Song et al. proposed the deep path-controlled unfolding network (DPC-DUN) (Song, Chen, and Zhang 2023), which employs a path-control selector to determine the optimal path for each image, striking a balance between performance and complexity. More notable instances encompass DGUNet (Mou, Wang, and Zhang 2022), CSformer (Ye et al. 2023), OCTUF (Song et al. 2023), LTwIST (Gan et al. 2024), UFC-Net (Wang and Gan 2024), among others (Kong et al. 2024; Kulkarni et al. 2016; Metzler, Mousavi, and Baraniuk 2017; Wu, Rosca, and Lillicrap 2019; Zhang, Zhao, and Gao 2020; Shi et al. 2019; Gan, Guo, and Liu 2024; Guo and Gan 2024). Despite the remarkable performance and interpretability of these methods, they still struggle to precisely reconstruct intricate details under limited sampling conditions, frequently resulting in block artifacts.

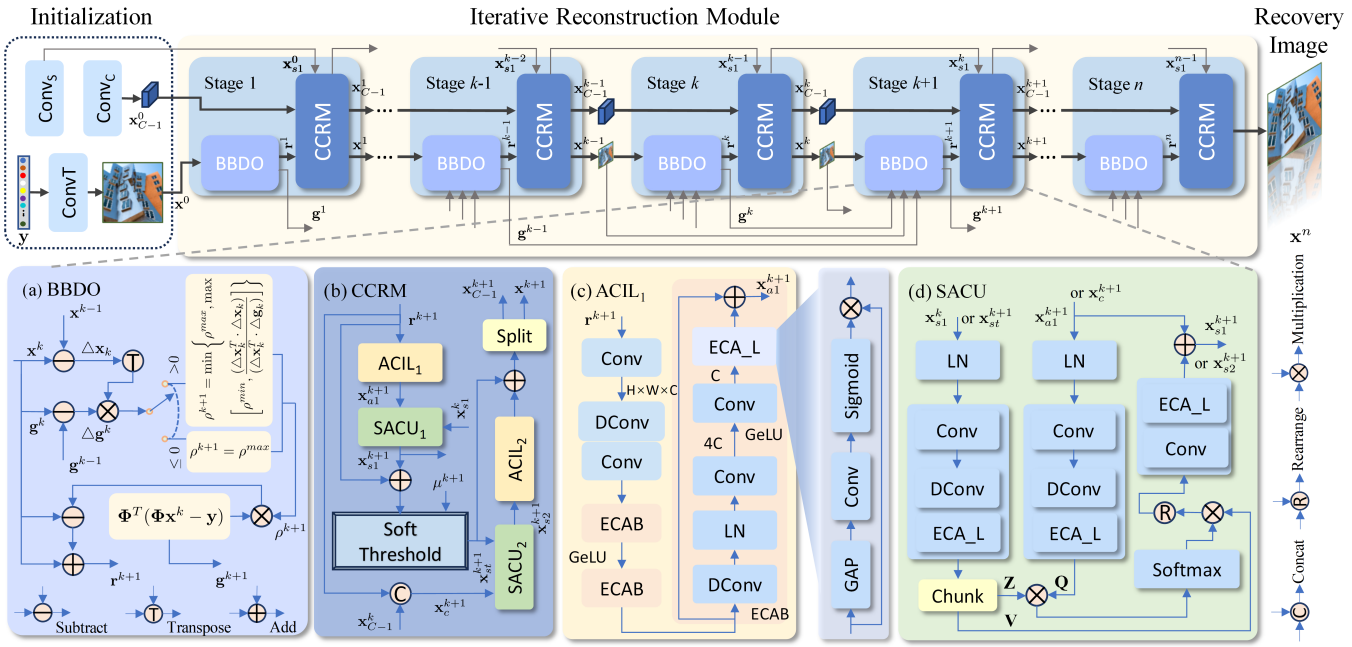


Figure 1: The overview of reconstuction module in our proposed SCU-Net. The iterative recovery module is composed of n recovery stages, each encompassing two principal modules: BBDO and CCRM. The subsequent layer further delineates the specific workflow of each iterative phase, and the operational functionalities of the specially devised ACIL and SACU.

Method

SCU-Net Architecture

Fig. 1 illustrates a detailed depiction of our proposed SCU-Net, highlighting the initialization and the iterative reconstruction module, providing an intuitive perspective on the intricate framework structure. To begin with, SCU-Net uses a convolutional layer with a kernel size of $M \times 1 \times \sqrt{N} \times \sqrt{N}$ as the trainable measurement matrix $\mathbf{\Phi}$ for a given sampling rate $\tau = \frac{M}{N}$, facilitating the sampling process on the original image $\mathbf{x}$, thereby obtaining the measurements denoted as $\mathbf{y}$. Following this, an upsampling convolution ConvT$(\cdot)$ meaning $\mathbf{\Phi}^T$ (the transpose of $\mathbf{\Phi}$) is performed on $\mathbf{y}$ to generate the initial image estimate $\mathbf{x}^0$, i.e., $\mathbf{x}^0 = \mathbf{\Phi}^T\mathbf{y} =$ ConvT$(\mathbf{y})$. Subsequently, $\mathbf{x}^0$ serves as input to the reconstruction network, which iteratively updates this estimate through a sequence of repeated reconstruction stages. After traversing through n stages, the estimate value $\mathbf{x}^n$ is obtained, signifying the ultimate reconstructed image. Within the iterative reconstruction module, each stage comprises two sub-modules: BB-gradient Descent Optimizer (BBDO) and Channel-guided Cross-attention Reconstruction Module (CCRM). These two sub-modules correspond to the gradient descent and proximal mapping steps in the SPGL1 iteration, respectively. The reconstruction module of the $(k+1)^{th} (0 \leq k < n)$ stage can be formulated as follows:

$$\begin{cases} (\mathbf{r}^{k+1}, \mathbf{g}^{k+1}) = \text{BBDO}(\mathbf{x}^{k-1}, \mathbf{x}^k, \mathbf{g}^{k-1}, \mathbf{g}^k), \\ (\mathbf{x}^{k+1}, \mathbf{x}_{C-1}^{k+1}, \mathbf{x}_{s1}^{k+1}) = \text{CCRM}(\mathbf{r}^{k+1}, \mathbf{x}_{s1}^k, \mathbf{x}_{C-1}^k), \end{cases} \tag{3}$$

where $\{\mathbf{x}^{k-1}, \mathbf{x}^k, \mathbf{g}^{k-1}, \mathbf{g}^k\}$ represent the inputs of BBDO, while $\{\mathbf{r}^{k+1}, \mathbf{g}^{k+1}\}$ denote the outputs of BBDO, and $\{\mathbf{r}^{k+1}, \mathbf{x}_{s1}^k, \mathbf{x}_{C-1}^k\}$ serve as the inputs of CCRM, with $\{\mathbf{x}^{k+1}, \mathbf{x}_{C-1}^{k+1}, \mathbf{x}_{s1}^{k+1}\}$ as the resulting outputs of CCRM. Moreover, only $\mathbf{x}^n$ of CCRM is necessary in the n^{th} phase.

BB-gradient Descent Optimizer

In each reconstruction phase, BBDO initially utilizes the BB step size update strategy to adjust the gradient descent stride for this iteration. Subsequently, the gradient descent method is employed to calculate the gradient of the data fidelity term, aiming to minimize the deviation between the original image and the estimate data. The complete procedure of BBDO in the $(k+1)^{th}$ stage can be mathematically formulated as:

$$\mathbf{r}^{k+1} = \mathbf{x}^k - \rho^{k+1}\mathbf{\Phi}^T(\mathbf{\Phi} \cdot \mathbf{x}^k - \mathbf{y}), \tag{4}$$

where $\mathbf{x}^k$ represents the output of the k^{th} stage, $\mathbf{r}^{k+1}$ signifies the gradient computation result of $\mathbf{x}^k$, and ρ^{k+1} denotes the gradient descent step size of the $(k+1)^{th}$ stage. Let $\mathbf{g}^{k+1} = \mathbf{\Phi}^T(\mathbf{\Phi} \cdot \mathbf{x}^k - \mathbf{y})$, the update rule of ρ^{k+1} depicted as follows:

$$\rho^{k+1} = \begin{cases} \min\{\rho^{max}, \max[\rho^{min}, \frac{\triangle \mathbf{x}_k^T \triangle \mathbf{x}_k}{\triangle \mathbf{x}_k^T \triangle \mathbf{g}_k}]\}, & \triangle \mathbf{x}_k^T \triangle \mathbf{g}_k > 0, \\ \rho^{max}, & \triangle \mathbf{x}_k^T \triangle \mathbf{g}_k \leq 0, \end{cases} \tag{5}$$

where $\triangle \mathbf{x}_k = \mathbf{x}^k - \mathbf{x}^{k-1}$, $\triangle \mathbf{g}_k = \mathbf{g}^k - \mathbf{g}^{k-1}$, $\triangle \mathbf{x}_k^T$ is the transpose of $\triangle \mathbf{x}_k$; $\mathbf{x}^{k-1}$ represents the output of the $(k-1)^{th}$ stage, $\mathbf{g}^{k-1}$ and $\mathbf{g}^k$ are the intermediate variables of BBDO in the $(k-1)^{th}$ and k^{th} stages, respectively. In particular, the gradient descent step size update begins from the second stage when $k = 1$, with the initial condition $\mathbf{g}^0 = \mathbf{x}^0$.

Datasets	Methods	CS Ratios					
		0.01	0.04	0.10	0.25	0.30	0.40
Set14	TransCS (TIP'22)	20.91/0.4853	25.50/0.7133	28.81/0.8343	33.37/0.9244	34.02/0.9349	36.68/0.9572
	DGUNet$^+$ (CVPR'22)	21.86/0.5409	25.88/0.7250	29.34/0.8455	33.70/0.9294	34.69/0.9412	36.78/0.9590
	AutoBCS (TCYB'23)	20.93/0.5343	25.07/0.7153	28.00/0.8286	32.14/0.9203	33.10/0.9308	34.94/0.9499
	CSformer (TIP'23)	21.89/0.5437	25.44/0.7023	28.34/0.8158	32.46/0.9143	-/-	-/-
	DPC-DUN (TIP'23)	17.17/0.3684	24.02/0.6487	28.03/0.7950	32.78/0.9023	33.92/0.9189	35.92/0.9430
	TCS-Net (TCI'23)	21.64/0.5219	25.25/0.7073	28.19/0.8283	32.23/0.9206	32.05/0.9194	33.58/0.9387
	LTwIST (TCSVT'24)	21.49/0.5190	25.47/0.7112	28.88/0.8352	33.42/0.9249	34.57/0.9382	36.61/0.9559
	MTC-CSNet (TCYB'24)	21.69/0.5312	25.20/0.7028	28.48/0.8336	32.64/0.9227	32.75/0.9285	33.91/0.9414
	UFC-Net (CVPR'24)	21.79/0.5324	25.67/0.7163	29.10/0.8363	33.81/0.9259	34.87/0.9390	36.93/0.9572
	SCU-Net (ours)	**22.09/0.5525**	**26.13/0.7313**	**29.55/0.8468**	**34.20/0.9318**	**35.37/0.9442**	**37.39/0.9607**
General100	TransCS (TIP'22)	21.66/0.5415	27.25/0.7843	31.39/0.8918	37.08/0.9604	37.81/0.9669	40.74/0.9806
	DGUNet$^+$ (CVPR'22)	22.86/0.6190	27.92/0.8078	32.41/0.9073	37.55/0.9645	38.87/0.9724	41.07/0.9821
	AutoBCS (TCYB'23)	22.24/0.6164	27.10/0.7964	30.76/0.8927	35.92/0.9581	36.72/0.9645	38.65/0.9757
	CSformer (TIP'23)	23.08/0.6321	27.26/0.7864	31.04/0.8824	36.17/0.9548	-/-	-/-
	DPC-DUN (TIP'23)	17.36/0.4181	26.13/0.7367	31.17/0.8716	36.50/0.9481	37.76/0.9590	39.95/0.9731
	TCS-Net (TCI'23)	22.58/0.5978	26.57/0.7712	29.90/0.8748	34.63/0.9504	35.11/0.9521	36.84/0.9636
	LTwIST (TCSVT'24)	22.69/0.5989	27.53/0.7935	31.91/0.8990	37.31/0.9616	38.61/0.9699	40.85/0.9806
	MTC-CSNet (TCYB'24)	22.97/0.6140	27.27/0.7855	31.33/0.8971	36.35/0.9597	36.54/0.9604	37.36/0.9683
	UFC-Net (CVPR'24)	23.08/0.6145	27.92/0.7988	32.31/0.9014	37.75/0.9624	38.89/0.9704	41.17/0.9810
	SCU-Net (ours)	**23.58/0.6410**	**28.48/0.8141**	**32.84/0.9099**	**38.25/0.9666**	**39.55/0.9741**	**41.66/0.9832**

Table 1: PSNR (dB) and SSIM comparisons of our proposed SCU-Net and nine competing methods on datasets Set14 and General100. Refer to *Sect. 3.2 of* ***Supp*** for additional comparison algorithms and datasets (McM18, BSD68, OST300, etc).

Channel-guided Cross-attention Reconstruction Module

Following the BBDO module, CCRM tackles the proximal mapping problem associated with nonlinear sparse variations in the SPGL1 algorithm. Assuming at the $(k+1)^{th}$ stage, CCRM takes $\mathbf{r}^{k+1}$, one of the outputs of BBDO, and the dual reinforcement variables $\mathbf{x}_{C-1}^{k}$ and $\mathbf{x}_{s1}^{k}$ as inputs, producing $\mathbf{x}^{k+1}$, $\mathbf{x}_{C-1}^{k+1}$ and $\mathbf{x}_{s1}^{k+1}$ as outputs. Particularly, during the first phase, i.e., $k=0$, inputs $\mathbf{x}_{s1}^{0}$ and $\mathbf{x}_{C-1}^{0}$ of CCRM are independently derived through two convolutional procedures $\text{Conv}_S(\cdot)$ and $\text{Conv}_C(\cdot)$, respectively. The procedure of CCRM can be delineated as follows:

$$\begin{aligned}\{\mathbf{x}^{k+1}, \mathbf{x}_{C-1}^{k+1}\} =& \mathcal{F}_{split}(\text{ACIL}_2(\text{SACU}_2(\mathcal{S}_{\mu^{k+1}}(\text{SACU}_1 \\ & (\text{ACIL}_1(\mathbf{r}^{k+1}), \mathbf{x}_{s1}^{k})), \mathbf{x}_{C-1}^{k}))),\end{aligned} \tag{6}$$

where μ^{k+1} is the soft thresholding parameter for the $(k+1)^{th}$ stage, which can be automatically updated through backpropagation. $\mathcal{S}_{\mu^{k+1}}(\cdot)$ denotes the soft threshold operation with the threshold of μ^{k+1}, $\mathcal{F}_{split}(\cdot)$ decomposes the input of C channels (default C = 32) into two outputs: one channel and C-1 channels, denoted as $\mathbf{x}^{k+1}$ and $\mathbf{x}_{C-1}^{k+1}$, respectively. $\mathbf{x}^{k+1}$ signifies the reconstructed image at the $(k+1)^{th}$ stage, while $\mathbf{x}_{C-1}^{k+1}$ stands for one of the enhanced variables produced in that stage.

Drawing inspiration from DL, the sparsity regularization problem is solved by a non-linear transformation function $\mathcal{H}(\cdot)$ instead of the manually customized operator $\mathbf{K}$ in Eq. (1). By substituting $\mathcal{H}(\cdot)$ for $\mathbf{K}$ in Eq. (1), the solution approach for the sparsity regularization problem is obtained:

$$\mathbf{x}^{k+1} = \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{x} - \mathbf{r}^{k+1}\|_{\ell_2}^2 + \lambda^{k+1}\|\mathcal{H}(\mathbf{x})\|_{\ell_1}. \tag{7}$$

According to Theorem 1 (Zhang and Ghanem 2018), we attain the following approximation:

$$\|\mathcal{H}(\mathbf{x}) - \mathcal{H}(\mathbf{r}^{k+1})\|_{\ell_2}^2 \approx \delta\|\mathbf{x} - \mathbf{r}^{k+1}\|_{\ell_2}^2, \tag{8}$$

where δ is a constant related to $\mathcal{H}(\cdot)$. By incorporating this relationship into Eq. (7), we derive the optimization problem as follows:

$$\mathbf{x}^{k+1} = \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathcal{H}(\mathbf{x}) - \mathcal{H}(\mathbf{r}^{k+1})\|_{\ell_2}^2 + \mu^{k+1}\|\mathcal{H}(\mathbf{x})\|_{\ell_1}, \tag{9}$$

where $\mu^{k+1} = \delta\lambda^{k+1}$ is accepted as a learnable parameter. Subsequently, we obtain the closed-form solution for $\mathcal{H}(\mathbf{x}^k)$:

$$\mathcal{H}(\mathbf{x}^{k+1}) = \mathcal{S}_{\mu^{k+1}}(\mathcal{H}(\mathbf{r}^{k+1})). \tag{10}$$

Let $\mathbf{x}_T^{k+1} = \mathcal{F}_{split}^{-1}(\mathbf{x}^{k+1}, \mathbf{x}_{C-1}^{k+1})$, where $\mathcal{F}_{split}^{-1}$ is the inverse of $\mathcal{F}_{split}$. Define function $\mathcal{H}$ as $\mathcal{H}(\cdot,\cdot) = \text{SACU}_1(\text{ACIL}_1(\cdot),\cdot)$. Let the left inverse of $\mathcal{H}(\cdot,\cdot)$, i.e., $\widetilde{\mathcal{H}}(\cdot,\cdot) = \text{ACIL}_2(\text{SACU}_2(\cdot,\cdot))$, thus yielding the optimal solution:

$$\begin{aligned}\mathbf{x}_T^{k+1} = \widetilde{\mathcal{H}}(\mathcal{S}_{\mu^{k+1}}(\mathcal{H}(\mathbf{r}^{k+1}, \mathbf{x}_{s1}^{k})), \mathbf{x}_{C-1}^{k}) = \text{ACIL}_2(\text{SACU}_2 \\ (\mathcal{S}_{\mu^{k+1}}(\text{SACU}_1(\text{ACIL}_1(\mathbf{r}^{k+1}), \mathbf{x}_{s1}^{k})), \mathbf{x}_{C-1}^{k})).\end{aligned} \tag{11}$$

By substituting the formulas of $\mathcal{H}(\cdot,\cdot)$ and $\widetilde{\mathcal{H}}(\cdot,\cdot)$, we can derive an equivalent relationship as shown in Eq. (11), which aligns with the input structure of function $\mathcal{F}_{split}(\cdot)$ in Eq. (6), thus demonstrating the rationality and convergence of our proposed method.

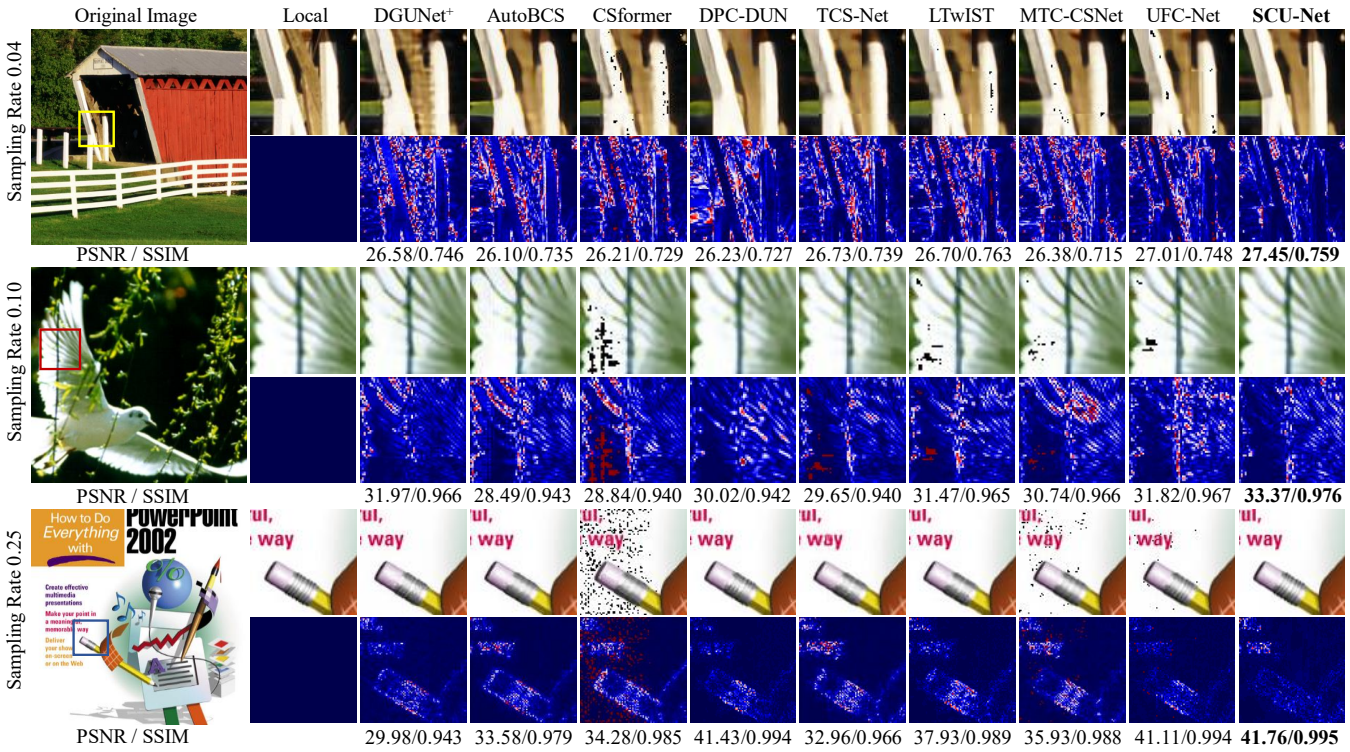


Figure 2: Visual comparisons of reconstructed images between SCU-Net and nine competing algorithms. The top row displays the comparison images in McM18 with $\tau = 0.04$. The second row corresponds to comparison images in the General100 dataset with $\tau = 0.10$, while the third row shows comparison images in Set14 with $\tau = 0.25$.

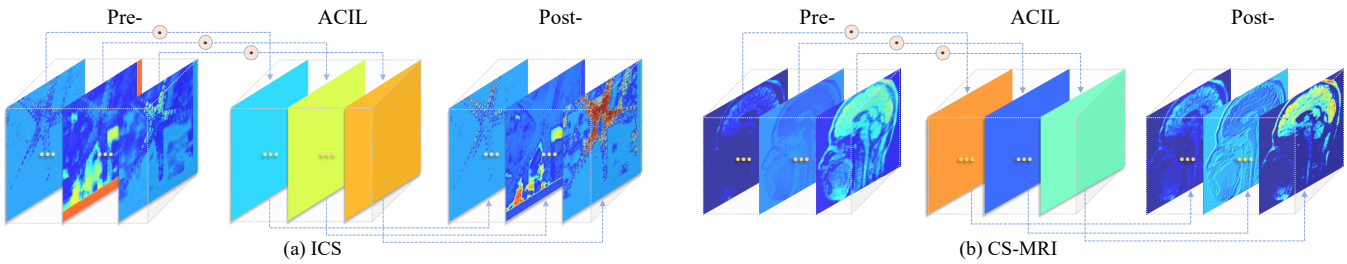
Adaptive Channel Interaction Layer. With the goal of enhancing the network's capacity to learn and represent intricate features of images more efficiently, we introduce ACIL, which focuses on improving convolutional feature extraction by amplifying the significance of channel dimensions through attention, extending the receptive field, and consequently enhancing the overall performance of the model. Notably, there are two instances of ACIL within CCRM: ACIL_1 and ACIL_2. ACIL_1 is elaborated specifically in Fig. 1(c), while ACIL_2 differs subtly from ACIL_1 as it represents the inverse process of ACIL_1 and includes layer normalization after the first efficient channel attention block (ECAB) (Wang et al. 2020). Therefore, a thorough explanation of ACIL_1 is adequate. To begin with, the feature map with a channel dimension of 1 is extended to C (number of channels) using a 1×1 convolution, followed by consecutive operations involving a 3×3 convolution ($\text{DConv}(\cdot)$) and another 1×1 convolution. Next, an ECAB block ($\text{ECAB}_1(\cdot)$) is utilized to adjust channel importance, enhancing feature representation and minimizing redundant information. The output of this process is then activated using the GeLU activation function and further processed by two ECAB blocks ($\text{ECAB}_2(\cdot)$ and $\text{ECAB}_3(\cdot)$) to obtain the module's output. The detailed operational procedure of ACIL_1 can be articulated as follows:

$$\begin{aligned}\mathbf{x}_{a1}^{k+1} =& \text{ECAB}_3(\text{ECAB}_2(\text{GeLU}(\text{ECAB}_1(\text{Conv}\\ &(\text{DConv}(\text{Conv}(\mathbf{r}^{k+1}))))))),\end{aligned} \tag{12}$$

where $\mathbf{x}_{a1}^{k+1}$ corresponds to the output of ACIL_1 in the $(k+1)^{th}$ stage.

Spatially Augmented Channel-aware Unit. To ensure maximal information transmission and emphasize the correlation among features, we develop SACU, which aims to enhance information flow, surpass the local inherent properties of pure convolutions, and strengthen region feature extraction while capturing both long-range dependencies and spatial relationships in images. Similarly, SACU also exhibits two instances, with an identical internal structure in each instance. As illustrated in Fig. 1(d), SACU_1 at the $(k+1)^{th}$ stage receives the output $\mathbf{x}_{s1}^{k}$ of SACU_1 at the k^{th} stage and $\mathbf{x}_{a1}^{k+1}$ as inputs and yields $\mathbf{x}_{s1}^{k+1}$ as the output. Initially, $\mathbf{x}_{s1}^{k}$ undergoes layer normalization ($\text{LN}(\cdot)$), followed by channel expansion from C to 2C through the convolution operation. Subsequently, it passes through the deep convolution and an ECA layer ($\text{ECA\_L}(\cdot)$, a structure inside ECAB) to obtain the output, which is then split into two parts along the channel dimension, forming tokens $\mathbf{Z}$ and $\mathbf{V}$.

Furthermore, $\mathbf{x}_{a1}^{k+1}$ undergoes layer normalization concurrently and then proceeds sequentially through a convolution, a deep convolution, and an ECA layer without altering the channel dimension, resulting in token $\mathbf{Q}$. The product of token $\mathbf{Q}$ with $\mathbf{Z}$, after passing through a softmax layer, is then multiplied by $\mathbf{V}$ to reset the dimensional parameters. Afterward, it undergoes a convolution and an ECA layer again before being added to $\mathbf{x}_{a1}^{k+1}$, one of the input tensors, to de-



(a) ICS (b) CS-MRI

Figure 3: Visualization analysis of channel attention weight distribution, with image CS on the left and CS-MRI on the right.

Datasets	τ	MTC-CSNet (TCYB 2024)		NesTD-Net (TIP 2024)		CPP-Net (CVPR2024)		SCU-Net	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
CIFAR10	0.01	14.31	0.2344	16.47	0.2823	15.61	0.2799	17.71	0.3145
	0.04	18.79	0.5760	21.04	0.6294	21.22	0.6266	21.54	0.6467
	Avg.	16.55	0.4052	18.76	0.4559	18.42	0.4533	19.63	0.4806
CIFAR100	0.01	14.53	0.2523	16.63	0.2993	15.81	0.2970	17.92	0.3301
	0.04	19.10	0.5863	21.45	0.6388	21.66	0.6370	22.03	0.6468
	Avg.	16.82	0.4193	19.04	0.4691	18.74	0.4670	19.98	0.4885

Table 2: Comparisons with other methods on large datasets.

rive the final output $\mathbf{x}_{s1}^{k+1}$ of SACU_1 at the $(k+1)^{th}$ stage. The whole process can be formulated as follows:

$$\begin{cases} (\mathbf{Z}, \mathbf{V}) = \mathcal{F}_{chunk}(\text{ECA\_L}(\text{Dconv}(\text{Conv}(\text{LN}(\mathbf{x}_{s1}^{k}))))), \\ \mathbf{Q} = \text{ECA\_L}(\text{Dconv}(\text{Conv}(\text{LN}(\mathbf{x}_{a1}^{k+1})))), \\ \mathbf{x}_{s1}^{k+1} = \mathbf{x}_{a1}^{k+1} + \text{ECA\_L}(\text{Conv}(\mathcal{F}_r(\mathbf{V} \cdot \text{Softmax}(\mathbf{Z} \cdot \mathbf{Q})))), \end{cases} \tag{13}$$

where $\mathcal{F}_{chunk}(\cdot)$ represents a function that divides the input into two equal parts along the channel dimension. Specifically, SACU_2 after the soft thresholding takes $\mathbf{x}_{st}^{k+1}$ and $\mathbf{x}_{c}^{k+1}$ as inputs and marks the output as $\mathbf{x}_{s2}^{k+1}$, while the rest of the process remains consistent. Here, $\mathbf{x}_{st}^{k+1}$ represents the output of soft thresholding at the $(k+1)^{th}$ stage, and $\mathbf{x}_{c}^{k+1} = \mathcal{F}_{cat}(\mathbf{r}^{k+1}, \mathbf{x}_{C-1}^{k})$, where $\mathcal{F}_{cat}(\cdot,\cdot)$ is a function that concatenates two inputs along the channel dimension.

Loss Function

The Mean Square Error (MSE) is utilized as the loss function, with $\mathbf{\Theta}$ encapsulating all trainable parameters. The loss function is defined as:

$$\mathcal{L}(\mathbf{\Theta}) = \frac{1}{2\mathcal{N}} \sum_{i=1}^{\mathcal{N}} \left\| \mathbf{x}^{(i)} - \text{SCU-Net}(\mathbf{x}^{(i)}, \mathbf{\Theta}) \right\|_{\ell_2}^2, \tag{14}$$

where $\mathbf{x}^{(i)}$ denotes the representation of the i^{th} original image in the training set $\{\mathbf{x}^{(i)}\}_{i=1}^{\mathcal{N}}$, $\mathcal{N}$ represents the total number of images in the training set, $\mathbf{\Theta} = \{\mathbf{\Phi}, \mathbf{\Phi}^T \rho^{1\sim n}, \mu^{1\sim n}\} \cup \{\mathcal{I}_j^{k+1}\}_{k=0}^{n-1}$, where $j \in [1, 2, \cdots, J]$ is the index for the filters. The backpropagation optimization for all parameters is detailed in *Sect. 2 of* ***Supp***.

Experiments

Refer to *Sect. 3.1 of* ***Supp*** for all experimental settings. Our source code is available at https://github.com/Charlene-12/SCU-Net.

Variants	BBDO	ACIL	SACU	ECAB	ECA_L	PSNR	SSIM	LPIPS
net1	✓	✗	✗	✗	✗	31.57	0.9252	0.0495
net2	✓	✓	✗	✗	✗	34.31	0.9494	0.0376
net3	✓	✓	✗	✓	✗	35.49	0.9573	0.0361
net4	✓	✗	✓	✗	✗	34.84	0.9523	0.0343
net5	✓	✗	✓	✗	✓	35.03	0.9543	0.0343
net6	✗	✓	✓	✗	✓	34.25	0.9482	0.0415
SCU-Net	✓	✓	✓	✓	✓	36.05	0.9606	0.0326

Table 3: SCU-Net ablative experiments with distinct modules on the Set11 dataset when $\tau = 0.25$.

Comparisons

As illustrated in Tab. 1, our proposed SCU-Net is evaluated against nine distinct algorithms across popular benchmark datasets such as Set14 (Zeyde, Elad, and Protter 2012) and General100 (Dong, Loy, and Tang 2016), under six diverse sampling rates $\tau \in \{0.01, 0.04, 0.10, 0.25, 0.30, 0.40\}$. The comparison algorithms include TransCS (Shen et al. 2022), DGUNet$^+$ (Mou, Wang, and Zhang 2022), AutoBCS (Gan et al. 2023a), CSformer (Ye et al. 2023), DPC-DUN (Song, Chen, and Zhang 2023), TCS-Net (Gan et al. 2023b), LTwIST (Gan et al. 2024), MTC-CSNet (Shen et al. 2024) and UFC-Net (Wang and Gan 2024). The table highlights superior results in red boldface and suboptimal outcomes in blue. It is obviously that the proposed SCU-Net exhibits exceptional performance. Taking the General100 dataset as an example, across seven different sampling rates, our SCU-Net shows a remarkable improvement in PSNR compared to the second-best algorithms, with enhancements (percentage gains) of 0.50 dB ($\sim$2.17%), 0.56 dB ($\sim$2.01%), 0.43 dB ($\sim$1.33%), 0.50 dB ($\sim$1.32%), 0.66 dB ($\sim$1.70%) and 0.49 dB ($\sim$1.19%), respectively. We further compare SCU-Net with MTC-CSNet, NesTD-Net (Gan, Guo, and Liu 2024) and CPP-Net (Guo and Gan 2024) in Tab. 2 on two extensive datasets, CIFAR10 and CIFAR100 (Krizhevsky, Sutskever, and Hinton 2017). Our SCU-Net outperforms these counterparts, particularly at a low sampling rate of 0.01, where its PSNR is 2.10 dB ($\sim$13.35%) higher than that of CPP-Net.

Furthermore, Fig. 2 illustrates the comparisons between reconstruction images generated by our SCU-Net and the aforementioned nine algorithms. To facilitate observation, we intentionally provide enlarged views of selected regions along with their corresponding error maps. The results demonstrate that the proposed SCU-Net possesses the capability to produce high-quality reconstructed images, which exhibit rich details, crisper and smooth lines, and reduced block artifacts. *Sect. 3.2 of* ***Supp*** presents a more comprehensive comparison of our SCU-Net with other algorithms.

Methods	fastMRI						Brain					
	0.05		0.1		0.15		0.05		0.1		0.2	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Zero-filled (JMRI'01)	23.23	0.4564	25.38	0.5314	26.31	0.5763	24.20	0.5417	26.82	0.6031	30.42	0.7230
ISTA-Net$^+$ (CVPR'18)	27.02	0.5498	27.20	0.5946	28.85	0.6922	31.28	0.8547	34.62	0.9035	38.57	0.9478
DC-CNN (TMI'17)	26.81	0.5432	27.07	0.5899	27.92	0.6624	30.81	0.8370	34.45	0.8985	38.41	0.9466
RDN (AAAI'18)	27.00	0.5471	27.32	0.5967	28.89	0.6911	30.95	0.8422	34.38	0.8998	38.47	0.9474
CDDN (NIPS'19)	27.13	0.5535	27.42	0.6015	28.93	0.6941	31.58	0.8513	34.66	0.9013	38.64	0.9475
ADMM-CSNet (TPAMI'20)	27.06	0.5512	27.43	0.5999	28.79	0.6898	31.39	0.8610	34.47	0.8987	38.54	0.9473
HiTDUN (JSTSP'22)	27.08	0.5513	27.62	0.6057	29.10	0.6977	32.72	0.8770	35.34	0.9104	39.02	0.9509
PUERT (JSTSP'22)	26.93	0.5465	27.53	0.6043	28.51	0.6815	31.52	0.8542	34.84	0.9048	38.78	0.9495
LTwIST (TCSVT'24)	27.23	0.5517	27.70	0.6003	29.03	0.6914	31.30	0.8536	34.11	0.9043	36.68	0.9361
UFC-Net (CVPR'24)	27.43	0.5642	28.01	0.6202	29.46	0.7098	33.13	0.8871	35.66	0.9195	39.05	0.9518
SCU-Net (ours)	**27.64**	**0.5722**	**28.27**	**0.6285**	**29.60**	**0.7141**	**33.76**	**0.8946**	**36.12**	**0.9226**	**39.56**	**0.9544**

Table 4: PSNR (dB) and SSIM comparisons between SCU-Net and ten CS-MRI methods on fastMRI and Brain datasets.

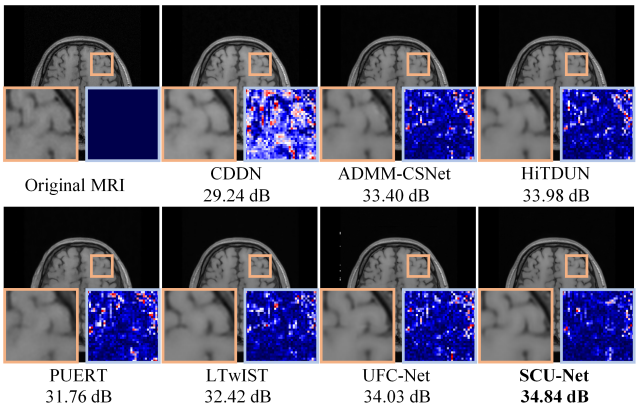


Figure 4: Comparisons of SCU-Net and other CS-MRI methods on magnetic resonance images.

Ablation Study

SCU-Net with different components. To validate the effectiveness of our designed module, we derive multiple variants, as depicted in Tab. 3, where cases including the module are indicated by a checkmark, while those not including it are marked with a cross. At $\tau = 0.25$, we perform diverse evaluation metrics comparison of SCU-Net with these variants on the Set11 dataset. It is noteworthy that we disaggregate ACIL and ECAB, and only variants that fully incorporate both are equivalent to the ACIL depicted in Fig. 1. A similar principle applies to SACU and ECA_L. The conclusion indicates that each module exhibits excellent functional efficacy, contributing significantly to the heightened performance level of the entire network. Concretely, net3 exhibits a PSNR improvement of 1.18 dB relative to net2, while net5 shows an enhancement of 0.19 dB compared to net4, which indicate that introducing attention mechanisms in the channel dimension facilitates a more refined tuning of feature weights and optimizes the model's ability to capture image structural information. Furthermore, the feature map comparisons in *Fig. 2 of* ***Supp*** highlight ACIU and SACU's enhanced fidelity for fine detail and structural preservation.

Visual analysis of channel weight distribution. We visualize the characteristic channels before and after ACIL and the channel weight distribution inside ACIL in image CS and CS-MRI respectively, as shown in Fig. 3. Comparing the pre- and post-ACIL feature maps demonstrates the efficient information filtering and feature enhancement achieved through weight adjustment of feature channels. Moreover, *Sect. 3.4 of* ***Supp*** details ablation studies of SCU-Net at different iteration stages, while *Sect. 3.5 of* ***Supp*** presents comparative reconstruction performance against four baseline methods under varying noise conditions. Complexity analysis is provided in *Sect. 3.6 of* ***Supp***.

SCU-Net for CS-MRI. We expand the proposed SCU-Net from image CS to the realm of magnetic resonance imaging, aiming to improve its generalizability and practical applicability. Here, we define the sampling matrix as $\mathbf{\Phi} = \mathbf{M} \cdot \mathbf{F}$, where $\mathbf{M}$ denotes a specific sampling mask and $\mathbf{F}$ signifies the discrete Fourier transform. Detailed experimental configurations are outlined in *Sect. 3.1 of* ***Supp***. We retrain our proposed SCU-Net model along with ten other methods (Zero-filled (Bernstein, Fain, and Riederer 2001), ISTA-Net$^+$ (Zhang and Ghanem 2018), DC-CNN (Schlemper et al. 2017), RDN (Sun et al. 2018), CDDN (Zheng, Fang, and Zhang 2019), ADMM-CSNet (Yang et al. 2018), HiTDUN (Zhang et al. 2022), PUERT (Xie et al. 2022), LTwIST (Gan et al. 2024), and UFC-Net (Wang and Gan 2024)) on specific fastMRI and Brain datasets and conduct a comparative analysis of the results. Notably, our SCU-Net consistently outperforms other methods in all scenarios. Fig. 4 is the reconstructed images on the brain dataset, showcasing magnified regions and corresponding error maps, which further illustrates this superior performance, emphasizing the SCU-Net's capacity to capture richer details and deliver superior image quality compared to alternative methods. Refer to *Sect. 3.8 of* ***Supp*** for fastMRI recovery images.

Conclusion

In this paper, we propose SCU-Net for CS, an innovative deep unrolling framework inspired by SPGL1. Refer to *Sect. 4 of* ***Supp*** for future works and limitations.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant 62471395, and in part by Basic Research Programs of Taicang under Grant TC2024JC17.

References

Barzilai, J.; and Borwein, J. M. 1988. Two-point step size gradient methods. *SIMA Journal of Numerical Analysis*, 8(1): 141–148.

Beck, A.; and Teboulle, M. 2009. A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems. *SIAM Journal on Imaging Sciences*, 2(1): 183–202.

Bernstein, M. A.; Fain, S. B.; and Riederer, S. J. 2001. Effect of windowing and zero-filled reconstruction of MRI data on spatial resolution and acquisition strategy. *Journal of Magnetic Resonance Imaging*, 14(3): 270–280.

Bioucas-Dias, J. M.; and Figueiredo, M. A. 2007. A new TwIST: Two-step iterative shrinkage/thresholding algorithms for image restoration. *IEEE Transactions on Image Processing*, 16(12): 2992–3004.

Borgerding, M.; Schniter, P.; and Rangan, S. 2017. AMP-Inspired Deep Networks for Sparse Linear Inverse Problems. *IEEE Transactions on Signal Processing*, 65(16): 4293–4308.

Boyd, S.; Parikh, N.; Chu, E.; Peleato, B.; Eckstein, J.; et al. 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1): 1–122.

Cheng, Z.; Chen, B.; Lu, R.; Wang, Z.; Zhang, H.; Meng, Z.; and Yuan, X. 2023. Recurrent Neural Networks for Snapshot Compressive Imaging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2): 2264–2281.

Dong, C.; Loy, C. C.; and Tang, X. 2016. Accelerating the super-resolution convolutional neural network. In *European Conference on Computer Vision (ECCV)*, 391–407.

Dong, Y.; Liu, Q.; Du, B.; and Zhang, L. 2022. Weighted Feature Fusion of Convolutional Neural Network and Graph Attention Network for Hyperspectral Image Classification. *IEEE Transactions on Image Processing*, 31: 1559–1572.

Duarte, M. F.; and Baraniuk, R. G. 2012. Kronecker Compressive Sensing. *IEEE Transactions on Image Processing*, 21(2): 494–504.

Gan, H.; Gao, Y.; Liu, C.; Chen, H.; Zhang, T.; and Liu, F. 2023a. AutoBCS: Block-based image compressive sensing with data-driven acquisition and noniterative reconstruction. *IEEE Transactions on Cybernetics*, 53(4): 2558–2571.

Gan, H.; Guo, Z.; and Liu, F. 2024. NesTD-Net: Deep NESTA-Inspired Unfolding Network With Dual-Path Deblocking Structure for Image Compressive Sensing. *IEEE Transactions on Image Processing*, 33: 1923–1937.

Gan, H.; Shen, M.; Hua, Y.; Ma, C.; and Zhang, T. 2023b. From patch to pixel: A transformer-based hierarchical framework for compressive image sensing. *IEEE Transactions on Computational Imaging*, 9: 133–146.

Gan, H.; Wang, X.; He, L.; and Liu, J. 2024. Learned Two-Step Iterative Shrinkage Thresholding Algorithm for Deep Compressive Sensing. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(5): 3943–3956.

Guo, Z.; and Gan, H. 2024. CPP-Net: Embracing Multi-Scale Feature Fusion into Deep Unfolding CP-PPA Network for Compressive Sensing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jalal, A.; Arvinte, M.; Daras, G.; Price, E.; Dimakis, A. G.; and Tamir, J. 2021. Robust Compressed Sensing MRI with Deep Generative Priors. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems*, volume 34, 14938–14954.

Kong, X.; Chen, Y.; Zheng, F.; and He, Z. 2024. Block Image Compressive Sensing with Local and Global Information Interaction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 2822–2830.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2017. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6): 84–90.

Kulkarni, K.; Lohit, S.; Turaga, P.; Kerviche, R.; and Ashok, A. 2016. ReconNet: Non-iterative reconstruction of images from compressively sensed measurements. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 449–458.

Meng, X.; and Kabashima, Y. 2024. QCS-SGM+: Improved Quantized Compressed Sensing with Score-Based Generative Models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 14341–14349.

Metzler, C.; Mousavi, A.; and Baraniuk, R. 2017. Learned D-AMP: Principled neural network based compressive image recovery. *Advances in Neural Information Processing Systems*, 30.

Mou, C.; Wang, Q.; and Zhang, J. 2022. Deep generalized unfolding networks for image restoration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 17399–17410.

Schlemper, J.; Caballero, J.; Hajnal, J. V.; Price, A. N.; and Rueckert, D. 2017. A deep cascade of convolutional neural networks for dynamic MR image reconstruction. *IEEE Transactions on Medical Imaging*, 37(2): 491–503.

Shen, M.; Gan, H.; Ma, C.; Ning, C.; Li, H.; and Liu, F. 2024. MTC-CSNet: Marrying Transformer and Convolution for Image Compressed Sensing. *IEEE Transactions on Cybernetics*, 1–13.

Shen, M.; Gan, H.; Ning, C.; Hua, Y.; and Zhang, T. 2022. TransCS: a transformer-based hybrid architecture for image compressed sensing. *IEEE Transactions on Image Processing*, 31: 6991–7005.

Shi, W.; Jiang, F.; Liu, S.; and Zhao, D. 2019. Image compressed sensing using convolutional neural network. *IEEE Transactions on Image Processing*, 29: 375–388.

Song, H.; Gong, J.; Meng, H.; and Lai, Y. 2024. Multi-Cross Sampling and Frequency-Division Reconstruction for Image Compressed Sensing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 4909–4917.

Song, J.; Chen, B.; and Zhang, J. 2023. Dynamic path-controllable deep unfolding network for compressive sensing. *IEEE Transactions on Image Processing*, 32: 2202–2214.

Song, J.; Mou, C.; Wang, S.; Ma, S.; and Zhang, J. 2023. Optimization-inspired cross-attention transformer for compressive sensing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6174–6184.

Sun, L.; Fan, Z.; Huang, Y.; Ding, X.; and Paisley, J. 2018. Compressed sensing MRI using a recursive dilated network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

van den Berg, E.; and Friedlander, M. P. 2009. Probing the Pareto Frontier for Basis Pursuit Solutions. *SIAM Journal on Scientific Computing*, 31(2): 890–912.

Wang, L.; Cao, M.; Zhong, Y.; and Yuan, X. 2023. Spatial-Temporal Transformer for Video Snapshot Compressive Imaging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(7): 9072–9089.

Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; and Hu, Q. 2020. ECA-Net: Efficient channel attention for deep convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 11534–11542.

Wang, X.; and Gan, H. 2024. UFC-Net: Unrolling Fixed-point Continuous Network for Deep Compressive Sensing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Wu, Y.; Rosca, M.; and Lillicrap, T. 2019. Deep compressed sensing. In *International Conference on Machine Learning*, 6850–6860.

Xie, J.; Zhang, J.; Zhang, Y.; and Ji, X. 2022. PUERT: Probabilistic under-sampling and explicable reconstruction network for CS-MRI. *IEEE Journal of Selected Topics in Signal Processing*, 16(4): 737–749.

Yang, Y.; Sun, J.; Li, H.; and Xu, Z. 2018. ADMM-CSNet: A deep learning approach for image compressive sensing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(3): 521–538.

Ye, D.; Ni, Z.; Wang, H.; Zhang, J.; Wang, S.; and Kwong, S. 2023. CSformer: Bridging convolution and transformer for compressive sensing. *IEEE Transactions on Image Processing*, 32: 2827–2842.

Zeyde, R.; Elad, M.; and Protter, M. 2012. On single image scale-up using sparse-representations. In *International Conference Curves and Surfaces*, 711–730.

Zhang, J.; and Ghanem, B. 2018. ISTA-Net: Interpretable optimization-inspired deep network for image compressive sensing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1828–1837.

Zhang, J.; Zhang, Z.; Xie, J.; and Zhang, Y. 2022. High-throughput deep unfolding network for compressive sensing MRI. *IEEE Journal of Selected Topics in Signal Processing*, 16(4): 750–761.

Zhang, J.; Zhao, C.; and Gao, W. 2020. Optimization-inspired compact deep compressive sensing. *IEEE Journal of Selected Topics in Signal Processing*, 14(4): 765–774.

Zhang, Y.; Shi, X.; Li, D.; Wang, X.; Wang, J.; and Li, H. 2023. A Unified Conditional Framework for Diffusion-based Image Restoration. In *Advances in Neural Information Processing Systems*, volume 36, 49703–49714.

Zhang, Z.; Liu, Y.; Liu, J.; Wen, F.; and Zhu, C. 2020. AMP-Net: Denoising-based deep unfolding for compressive image sensing. *IEEE Transactions on Image Processing*, 30: 1487–1500.

Zheng, H.; Fang, F.; and Zhang, G. 2019. Cascaded dilated dense network with two-step data consistency for MRI reconstruction. *Advances in Neural Information Processing Systems*, 32.